



OpenATE Inc.
The Open Solution for IC Tester

QSPI User Manual



Revision History

The following lists the additions, deletions and modifications in this manual at each revision.

| Date | Version | Remarks |
|--------------|---------|----------------------------|
| Aug 08, 2011 | 1.0 | Release the document |
| Aug 23,2011 | 1.0 | Add cal_reset |
| Aug 31,2011 | 1.1 | Add ext sense/force |
| Nov 23,2011 | 1.1 | Add rd_peid |
| Nov 25,2011 | 1.1 | Rename rd_peid to rd_pesno |
| Dec 28,2011 | 1.1 | add cal auto save/load |



Table of Contents

| No | Description | Page |
|------|---------------------------------|------|
| 1 | Overview | 4 |
| 2 | Block | 5 |
| 2.1 | Features | 5 |
| 2.2 | DC Specification | 6 |
| 2.3 | Digital Specification | 7 |
| 2.4 | Sequencer | 8 |
| 2.5 | Formater | 9 |
| 2.6 | Function Block | 10 |
| 2.7 | Local Memory Structure | 11 |
| 2.8 | Sequencer Structure | 12 |
| 2.9 | Formater Structure | 13 |
| 2.10 | Pin Eelectronics | 14 |
| 3 | Functions | 15 |
| 3.1 | Seqencer Functions | 15 |
| 3.2 | Formater Functions | 21 |
| 3.3 | Pin Electronics | 24 |
| 3.4 | Calibration | 28 |
| 3.5 | Counter Function | 32 |
| 3.6 | I2C Function | 33 |
| 3.7 | SPI Function | 34 |
| 4 | Pattern Format | 36 |
| 5 | Pin Out | 37 |
| | Figure1. QSPI Connector Pin Out | 39 |



OpenATE Inc.

The Open Solution for IC Tester

1 Overview

This document provides detailed description of qspi functions.

QSPI is a PXI 3U logic IC tester with very high density and performance. QSPI can offer the Test Solution Provider(TSP) the necessary functions to test quad sites digital channels when configuring a solution for a device. QSPI can not only a tester, but also a I2C controll card. Many QSPIs can be chained together to get higher site counts. QSPI can use PXI-TRIGGER to synchronize with other PXI instruments. QSPI offers intensive APIs in C language that can controll all the detailed hardware functions. TSP can use QSPI to build a customized test system or solution. Design engineers can use QSPI for design I2C device verification.



2 Block

2.1 Features

1. PXI Logic Tester with I2C controllers
2. Quad-Testers Per Board Architecture
3. PMU per pin, DPS per site
4. 10MHz Test Rate TG / RVS Per Pin
5. 8X4 BI-Direction Logic Test Pins
6. MAX. 256 Pin,32 Site Parallel Test
7. 64M Local Memory
8. 4 I2C controllers
9. 4 32 bits/10ns counters



2.2 DC Specification

1. RVS Per Pin -1 ~+10V 16Bit
2. PMU Per pin -1 ~ +10V 16Bit 32mA
3. 8 Current Range : 2UA,8UA,32UA,128UA,512UA,2MA,8 MA,32MA
4. DPS Per site -1 ~ +10V 16Bit 64~512mA



2.3 Digital Specification

1. TP MIN. 10MHz/ 10nS Resolution
2. TG Per Pin 10nS Resolution
3. Local Memory 64M
4. 1 Time Set



2.4 Sequencer

1. uCommand : FC,
2. FC Counter : 20 Bits
3. LMSYN to PXI TRIGGER 1~8
4. Start TRIGGER from PXI TRIGGER 1~8
5. Fail to PXI TRIGGER 1~8
6. FT Counter : 32 Bits
7. Cycle mode
8. Ignore Fail by LM Address

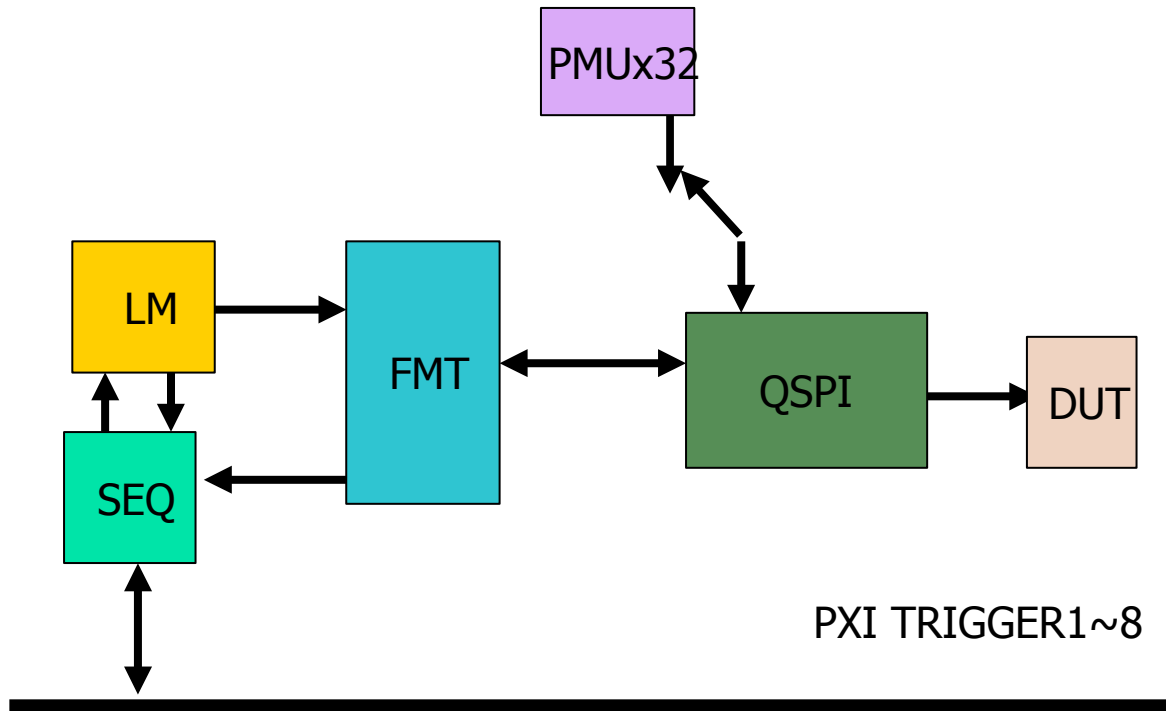


2.5 Formater

1. F,RZ,RO,SBC
2. Drive : 1,0,X,J,Q
3. Receive : H,L,Z
4. 1 Format Set
5. Window Strobe



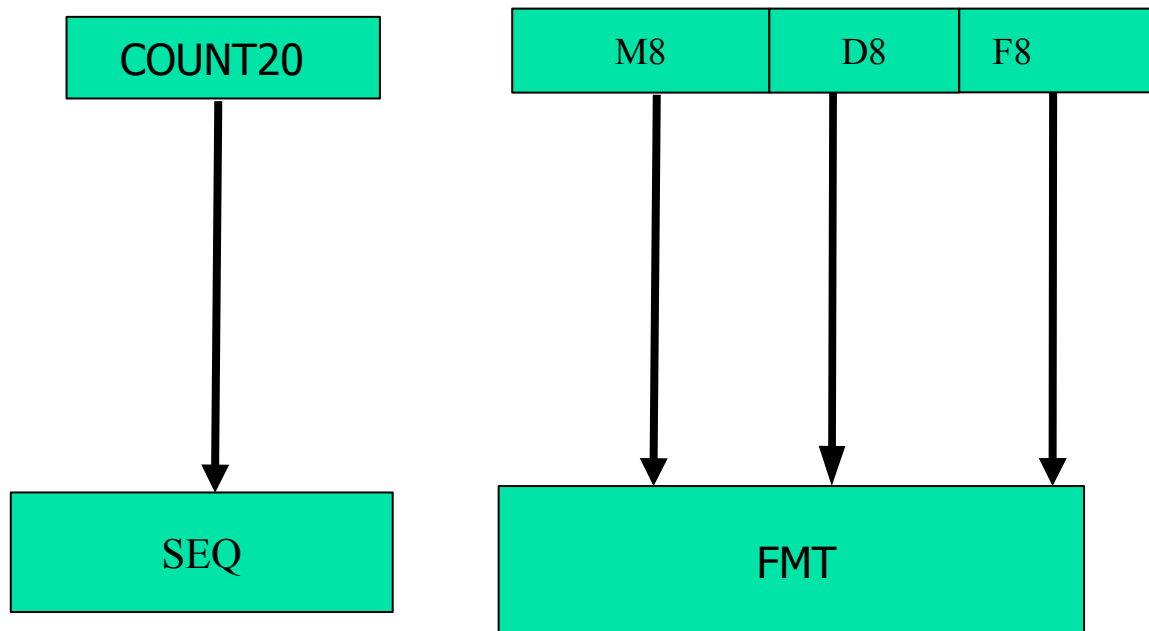
2.6 Function Block





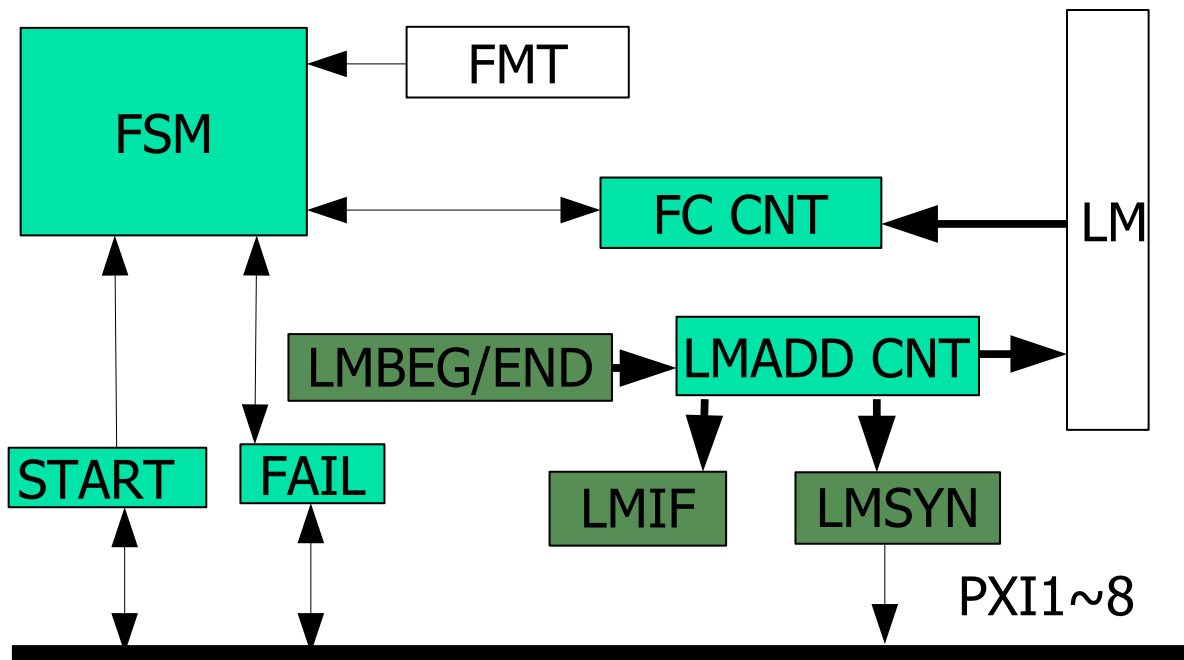
2.7 Local Memory Structure

20+24 Bit X (64 M) QSPI



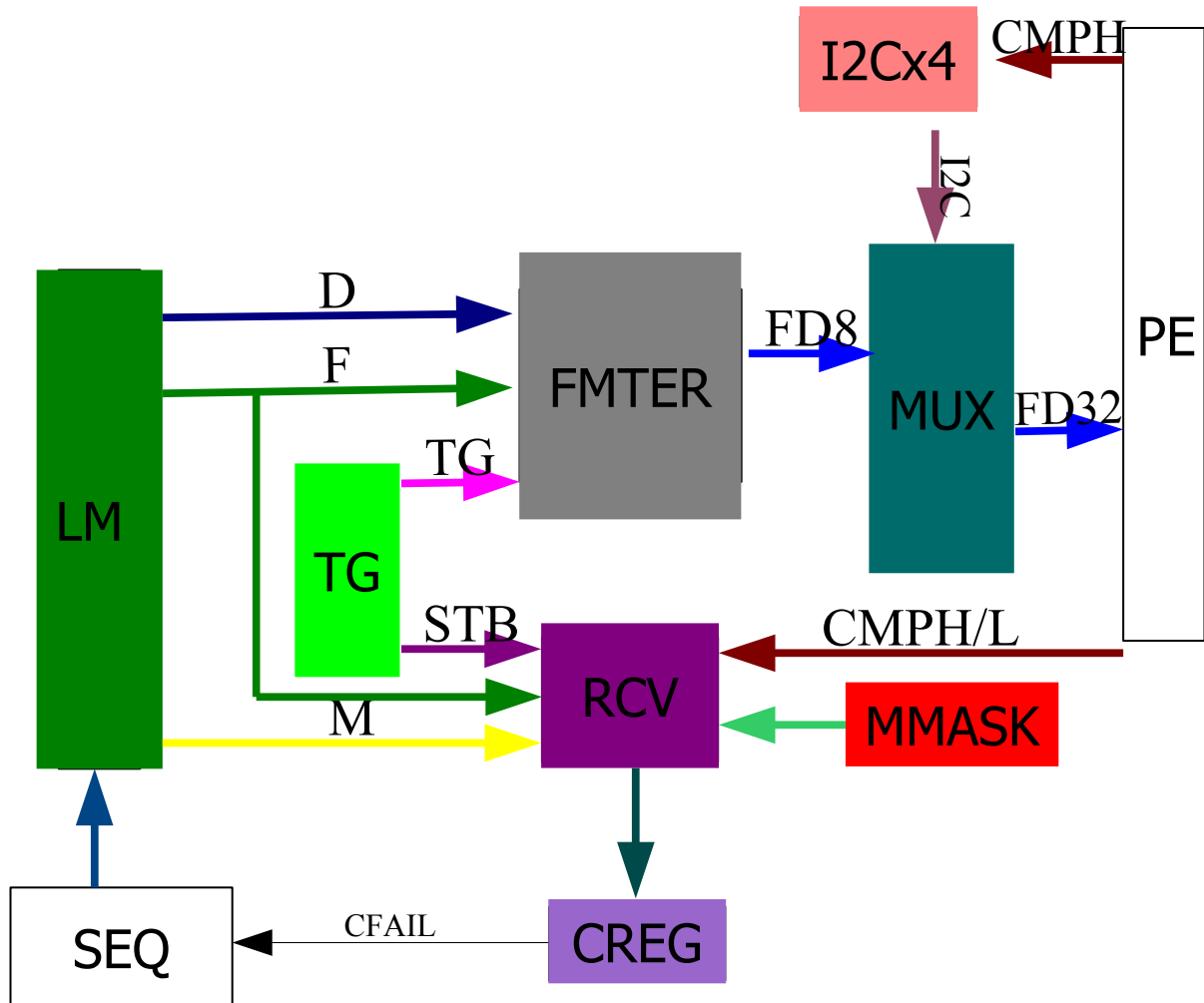


2.8 Sequencer Structure



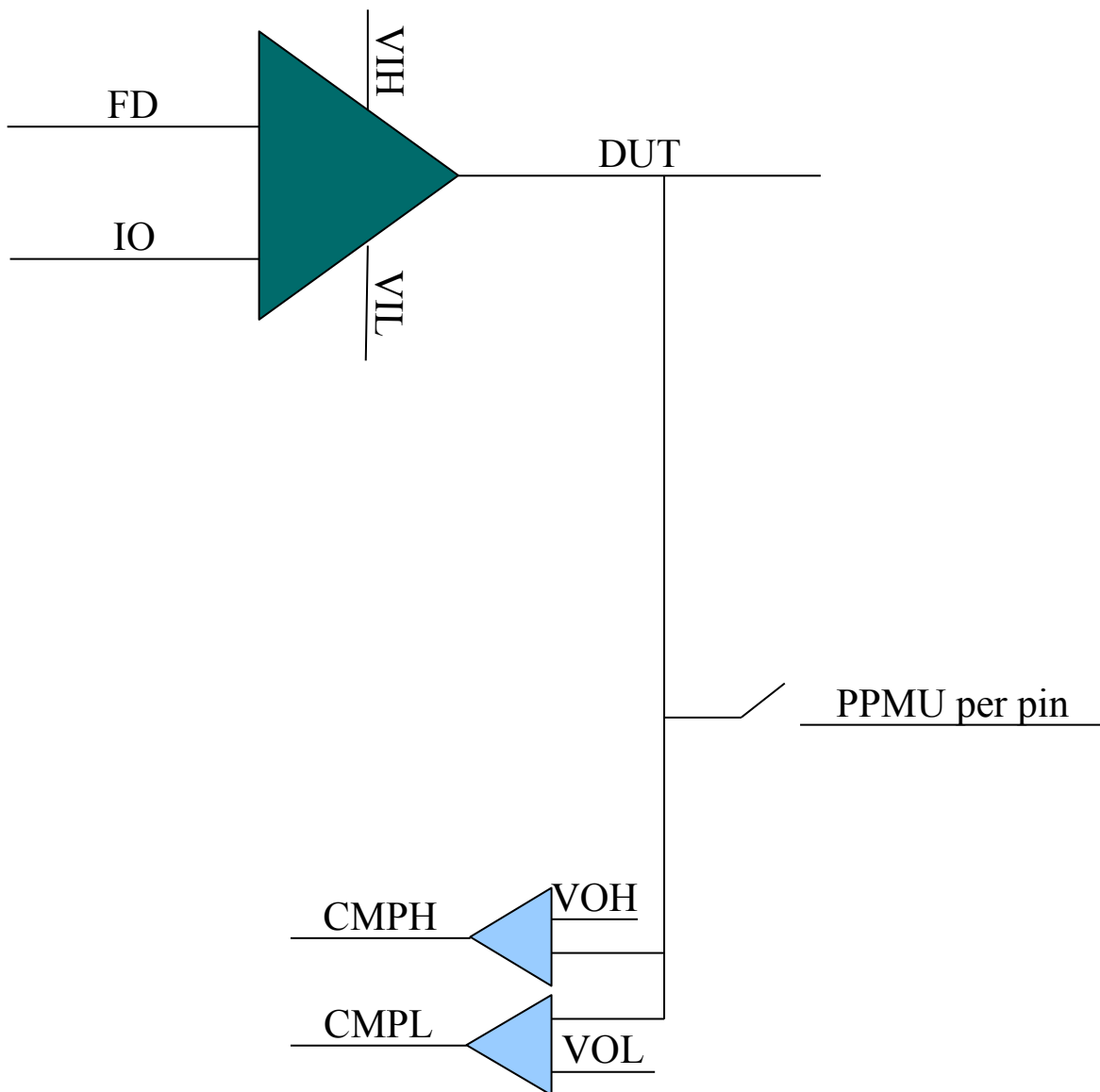


2.9 Formater Structure





2.10 Pin Electronics





3 Functions

QSPI API Functions:

3.1 Sequencer Functions

Definition

```
int qspi_init(void);
```

Return Value

board counts in system

Description

This function must be executed before any other functions.

If return value is 0, no qspi detected.

Definition

```
void qspi_reset(int bdno);
```

Description

reset sequencer and set FTCNT to 1 , CREG,ADDBEG,ADDEND, ADDSYN ADDIF to 0,MMSKto 0xFFFFFFFF

reset all DC level and DAC offset/gain to 0.0

if bdno=0, reset all boards

Definition

```
void qspi_set_ftcnt(int bdno, long cnt);
```

Description

set FTCNT reg(32bits) to cnt for board bdno,if bdno=0, set all boards

FTCNT is used to set loop test count.The pattern will be run from ADDBEG to ADDEND repeat FTCNT times.

Definition

```
void qspi_set_addbeg(int bdno, long add);
```

Description

set LMREG reg(27bits) to add for board bdno,if bdno=0, set all boards

LMADD will be loaded with LMREG;

Definition

```
void qspi_set_addend(int bdno, long add);
```

Description

set LMEND reg(27bits) to add for board bdno,if bdno=0, set all boards



Definition

void qspi_set_addif(int bdno, long add);

Description

set LMIF reg to add for board bdno, if bdno=0, set all boards when LMADD<= LMIF compare result will be ignored.

Definition

void qspi_set_addsyn(int bdno, long add);

Description

set LMSYN(27bits) reg to add for board bdno, if bdno=0, set all boards
LMSYN is used to generate LMSYNC signal which can be connect to PXI_TRIG by setting PXI reg or output to connector 1

Definition

void qspi_fstart(int bdno , int onoff);

Description

start/stop sequencer (run pattern) , if bdno=0, set all boards
Before using this function, LMBEG, LMEND has to be set to adequate value.
Set onoff to 0 to stop sequencer.

Definition

int qspi_check_tprun(int bdno);

Return Value

= 0 : sequencer stopped, = 1 : sequencer running

Description

check if sequencer running , bdno>0;
The sequencer stops in two conditions:
CFAIL detected from formater or PXI-TRIGGER bus.
LMEND reached and FT COUNTER count to the value defined by pe16_set_ftcnt(int bdno, long cnt);

Definition

int qspi_check_tpass(int bdno);

Return Value

= 1: no CFAIL detected by sequencer

Description

WHEN function test pass then TPASS=1, should be checked before setting tsatrt to 0, bdno>0;
pe16_fstart(bdno , 0) will reset TPASS;



APPLICATION:

User can make a user function for running pattern:

```
int FTEST(int bdn, unsigned long lbeg , unsigned long lend )
{
int rst;
qspi_set_checkmode(bdn, 0);
qspi_set_addbeg(bdn, lbeg);
qspi_set_addend(bdn, lend);
qspi_cycle(bdn, 0);
qspi_fstart(bdn, 1);
while (qspi_check_tprun(bdn) ; // wait for sequencer stop
rst=qspi_check_tpass(bdn);
qspi_fstart(bdn, 0);
return(rst);
}
```

This function can run pattern from lbeg to lend. If return value = 1, test pass;

Definition

```
void qspi_cycle(int bdno , int onoff);
```

Description

set CYCLE mode on/off, if bdno=0, set all boards

In CYCLE mode , sequencer will run from LMEBG to LMEND and back to LMBEG with unlimited times. Use qspi_fstart(bdno , 0) to stop sequencer. In CYCLE mode, any FAIL signal from receiver is ignored.

Definition

```
int qspi_check_sync(int bdno);
```

Return Value

LMSYNC bit

Description

LMSYNC=1 when LMSYN=LMADD , bdno>0;

Used for diagnosis

Definition

```
int qspi_check_testbeg(int bdno);
```

Return Value

TESTBEG bit

Description

return TESTBEG bit on CTRLRD reg, bdno>0;

Used for diagnosis



Definition

int qspi_check_ftend(int bdno);

Return Value

FTEND bit

Description

FTEND=1 when FT counter reach FTCNT set by qspi_set_ftcnt(int bdno, long cnt), bdno>0;

Used for diagnosis

Definition

int qspi_check_lend(int bdno);

Return Value

LEND bit

Description

LEND=1 when LMEND=LMADD,bdno>0;

Used for diagnosis

Definition

void qspi_set_pxi(int bdno, int data);

Description

set PXI reg(16bits) to data for board bdno,if bdno=0, set all boards

PXI reg is for controll of PXI_TRIG[0..7] BUS MSB and LSB of PXI reg will controll corespondent channel of PXI_TRIG BUS

MSB[n]LSB[n]="00" : not any connection to PXI_TRIG

MSB[n]LSB[n]="01" : output LMSYNC signal to PXI_TRIG[n] n=0..7

MSB[n]LSB[n]="10" : in/out for CFAIL signal

MSB[n]LSB[n]="11" : in/out use PXI_TRIG[n] as seqencer start channel (in and out)

when using PXI_TRIG the DIP SWITCH bit must be turn on on carrier board

Definition

void qspi_set_seq(int bdno, long data);

Description

set LM SEQ to data at LMADD for board bdno,if bdno=0, set all boards . Used for writing local memory

Definition

void qspi_set_lmf(int bdno, long data);

Description

set LM F to data at LMADD for board bdno,if bdno=0, set all boards . Used for writing local memory



Definition

long qspi_rd_seq(int bdno);

Return Value

LMSEQ

Description

return LM SEQ data at LMADD for board bdno. Used for checking local memory

Definition

long qspi_rd_lmf(int bdno);

Return Value

LMF

Description

return LM F data at LMADD for board bdno. Used for checking local memory

Definition

long qspi_rd_lmadd(int bdno);

Return Value

LMADD

Description

return LMADD for board bdno. Used for checking local memory

APPLICATION:

User can make a user function for checking pattern:

```
void CKLM(int bdn,unsigned long lbeg , unsigned long lend )
{
unsigned long j;
qspi_set_checkmode(bdn, 1);
for (j=lbeg;j<=lend;j++)
{
qspi_set_addbeg(bdn,j);
printf("BDN %d lmadd %d seq 0x%08lX lmf 0x%08lX \n",bdn,qspi_rd_lmadd(bdn),
qspi_rd_seq(bdn),qspi_rd_lmf(bdn));
}
qspi_set_checkmode(bdn, 0);
}
```

Definition

unsigned int qspi_rd_fcncnt(int bdno);

Return Value

FCCNT(20bit)

Description

FCCNT is set by FC in pattern. It counts down by every TP untill 0.

Used for diagnosis



Definition

unsigned int qspi_rd_ftcnt(int bdno);

Return Value

FT(32 bits)

Description

FT counter controll the loop counts the sequencer had run. Each LEND will increase the FT by one untill reaching FTCNT

Used for diagnosis

Definition

int qspi_check_checkmode(int bdno);

Return Value

CHECKMODE bit

Description

CHECKMODE set by qspi_set_checkmode(int bdno, int onoff), bdno>0;

Used for diagnosis

Definition

int qspi_check_dataready(int bdno);

Return Value

DATAREADY bit

Description

Each time when LMBEG changed, DATAREADY goes to 0 untill LM data updated from DDRIL.

DATAREADY =1 indicate LM data available, bdno>0

Used for diagnosis

Definition

long qspi_check_ucnt(int bdno);

Return Value

UCNT(32 bits)

Description

UCNT counts the TP steps sequencer has run, bdno>0;

Used for diagnosis

Definition

void qspi_set_checkmode(int bdno, int onoff);

Description

set checkmode bit for board bdno,if bdno=0, set all boards

CHECKMODE is used to set LM data accessed without passing through cache memory.This should be enabled only when LM cchecking is practicing. It can save time for LM read/ write. It is reseted to '0' when system reset. Only qspi_set_addbeg() is affected by this controll bit.



3.2 Formater Functions

Definition

```
void qspi_set_mmsk(int bdno, long data);
```

Description

set MMSK to data for board bdno,if bdno=0, set all boards after reset MMSK=0xFFFFFFFF when set to '0' will mask compare result for relative bit

Definition

```
void qspi_set_tp(int bdno,long data);
```

Description

set TP to data at for board bdno,if bdno=0, set all boards, tp is 16 bit counter . tp unit:10ns
Minimum is 10(10MHz) .

Definition

```
void qspi_set_tstrob(int bdno, int pno,long data);
```

Description

set TSTROB to data for board bdno port pno,if bdno=0, set all boards, pno=1..8,if pno=0, set all pins ,data is 16 bits,unit:10ns

Test fail will stop at next step (run over) when TP-TSTROB<60ns

Definition

```
void qspi_set_dstrob(int bdno, int pno,int data1,int data2);
```

Description

set window strobe to data1 and data2 for board bdno port pno,if bdno=0, set all boards,pno=1..8,if pno=0, set all pins ,data is 16 bits,unit:10ns.

When using qspi_set_tstrob, the default mode is edge strobe. If useing qspi_set_dstrob, the window strobe will be acitved. The compare strobe will be from data1 to data2. The window strobe can be set on per pin base.

Definition

```
void qspi_set_tstart(int bdno, int pno,long data);
```

Description

set TSTART to data for board bdno pin pno,if bdno=0, set all boards,pno=1..8,if pno=0, set all pins,data is 16 bits,unit:10ns



Definition

void qspi_set_tstop(int bdno, int pno, long data);

Description

set TSTOP to data at ts for board bdno pin pno, if bdno=0, set all boards, pno=1..8, if pno=0, set all pins, data is 16 bits, unit: 10ns

Definition

void qspi_set_rz(int bdno, long data);

Description

set RZ to data for board bdno, if bdno=0, set all boards, RZ is 16bit each bit map to one channel.

Definition

void qspi_set_ro(int bdno, long data);

Description

set RO to data for board bdno, if bdno=0, set all boards, RO is 16bit each bit map to one channel
OUTPUT format is defined by RZ and INV

RZ:RO=00 =>NRZ

RZ:RO=01 =>RTO

RZ:RO=10 =>RTZ

RZ:RO=11 =>SBC

Definition

long qspi_lmload(int bdno, long begadd, char* patternfile);

Return Value

last LMADD pattern loaded

Description

load pattern file to qspi board for board no bdno , begin at lmadd begadd .

For patternfile string, use "/" instead of "\".

Example : use "[c://OpenATE//patternfile](#)" for "c:\OpenATE\patternfile" .

Definition

unsigned int qspi_rd_cmph(int bdno);

Return Value

CMPH(32 bits), CMPH(1..8) for site 1, CMPH(9..16) for site 2.....

Description

CMPH(bit)= 1 when DUT >VOH



Definition

unsigned int qspi_rd_cmpl(int bdno);

Return Value

CMPL(32 bits)CMPL(1..8) for site 1, CMPL(9..16) for site 2.....

Description

CMPL(bit)=1 when DUT > VOL

Definition

unsigned int qspi_rd_creg(int bdno);

Return Value

CREG(32 bits)CREG(1..8) for site 1, CREG(9..16) for site 2.....

Description

CREG(bit) =1 when compare result failed



3.3 Pin Electronics

Definition

```
void qspi_set_vih(int bdno,int cno, int pno,double rv);
```

Description

set VIH to rv for board bdno pin pno,if bdno=0, set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

Definition

```
void qspi_set_vil(int bdno,int cno, int pno,double rv);
```

Description

set VIL to rv for board bdno pin pno,if bdno=0, set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

Definition

```
void qspi_set_voh(int bdno,int cno, int pno,double rv);
```

Description

set VOH to rv for board bdno pin pno,if bdno=0, set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

Definition

```
void qspi_set_vol(int bdno,int cno, int pno,double rv);
```

Description

set VOL to rv for board bdno pin pno,if bdno=0, set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

Definition

```
void qspi_pmufv(int bdno,int cno, int pno ,double rv,double clampi);
```

Description

set PMUFV to rv for board bdno pin cno ,if bdno=0, set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

set I clamp between clampi and -clampi. The value of cih and cil will define the I range automatically.

this command will change VIH

Definition

```
void qspi_pmufi(int bdno,int cno, int pno ,double ri,double cvh,double cvl);
```

Description

set PMUFI to ri mA for board bdno pin cno ,if bdno=0, set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

set V clamp between cvh and cvl. The value of ri will define the I range automatically.



Definition

void qspi_pmucv(int bdno,int cno, int pno ,double cvh,double cvl);

Description

set PMU compare V limit to cvh(PPMU-CH)/cvl(PPMU-CL) for board bdno pin pno ,if bdno=0, set all boards,set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins
this command will change VOH/VOL

Definition

void qspi_pmuci(int bdno,int cno, int pno ,double cih,double cil);

Description

set PMU compare I limit to cih/cil for board bdno pin pno ,if bdno=0, set all boards,set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins
this command will change VOH/VOL

Definition

void qspi_con_pmu(int bdno,int cno, int pno,int onoff);

Description

connect/disconnect CPMU to DUT for board bdno pin pno,if bdno=0, set all boards,set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

Definition

void qspi_con_esense(int bdno,int cno, int pno,int onoff);

Description

connect/disconnect EXT_SENSE to DUT for board bdno pin pno,if bdno=0, set all boards,set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

Definition

void qspi_con_eforce(int bdno,int cno, int pno,int onoff);

Description

connect/disconnect EXT_FORCE to DUT for board bdno pin pno,if bdno=0, set all boards,set all boards,cno=1..4,if cno=0, set all sites,pno=1..8,if pno=0, set all pins

Definition

int qspi_check_pmu(int bdno,int cno, int pno);

Return Value

1 when pass

Description

CPMU-CH/CL compare resut pass=1/fail=0 for board bdno pin pno
pass when CPMU-CL < DUT < CPMU-CH .



Definition

int qspi_pmuch(int bdno,int cno, int pno);

Return Value

1 when DUT > CPMU-CH

Description

CPMU-CH compare resut for board bdno pin pno

Definition

int qspi_pmucl(int bdno,int cno, int pno);

Return Value

1 when DUT > CPMU-CL

Description

CPMU-CL compare resut for board bdno pin pno

Definition

double qspi_vmeas(int bdno,int cno, int pno);

Return Value

return voltage value(V) of PMU for board bdno pin pno.

Description

Measure the real voltage value of PMU in FV or FI mode.

Definition

double qspi_imeas(int bdno,int cno, int pno);

Return Value

return current value(mA) of PMU for board bdno pin pno.

Description

Measure the real current loading of PMU in FV or FI mode.

Definition

void qspi_enbDPS(int bdno, int cno,int ganging);

Description

set PEMU32 to DPS mode for board bdno chip cno,if bdno=0, set all boards, cno=1..4,if cno=0, set all chips. Ganging=1..8. In DPS mode ,last pin in each chip(8,16,24,32) will chnaged to DPS. Each DPS can ganged with other pin to increase output current. Up to 8 pins can be ganged together to provide 8*64=512mA output current. When ganging=2, pin 8,7 will be ganged, ganging=3, pin 8,7,6 will be ganged. The ganged pin should be wired together and SENSE pin as well.



Definition

void qspi_disDPS(int bdno, int cno);

Description

Reset PEMU32 to PMU mode for board bdno chip cno,if bdno=0, set all boards, cno=1..4,if cno=0, set all chips.

Definition

void qspi_dpsfv(int bdno, int cno ,double rv,double clih, double cil);

Description

set DPSFV to rv for board bdno pin pno ,if bdno=0, set all boards, cno=1..4,if cno=0, set all chips.
set I clamp between clih and cil. The value of cih and cil will define the I range automaticly.

Definition

double qspi_dpismi(int bdno, int cno);

Return Value

return current value(mA) of DPS for board bdno chip cno.cno=1..4

Description

Measure the real current loading of DPS .



3.4 Calibration

To get an accurate DC level, calibration is necessary.

For all DC level offset adjustment, $vr = 0x0000(-5.4\% \text{ of FullScale } 16V/32mA) \sim 0x7FFF(0) \sim 0xFFFF(+5.4\% \text{ of FS})$

For DC level gain adjustment, $vr = 0x0000(0.875) \sim 0x7FFF(1.0) \sim 0xFFFF(1.125)$

The default offset and gain are set to $0x7FFF$ by `qspi_rst_pe`.

In general, most calibration points will be at 20% and 80% of the full scale value for that range.

$V_{mid} = 3.0$ for voltage level, $V_{mid} = 0.0$ for current level.

Level Calibration

Initialize

- Set Gain = 1.0; Offset = 0.0V($0x7FFF$)

Measure

- Set Level 1 = Cal Point 1. Measure Output1'
- Set Level 2 = Cal Point 2. Measure Output2'

Calculate

- $Gain' = (Output2' - Output1') / (Level2 - Level1)$
- $Offset' = (Output2' - V_{mid}) - Gain' \cdot (Level2 - V_{mid})$

Finish

- Set Offset = $- Offset' / Gain'$
- Set Gain = $1.0 / Gain'$

Definition

```
void qspi_set_Dacx(int bdno,int cno, int pno,int dac,double rv);
```

Description

set DAC to rv for board bdno pin pno,if bdno=0, set all boards,if pno=0, set all pins
dac=0..9
0:VTT,1:VIL,2:VOH,3:VOL,4:FV/VIH,5:FI,,6:VCLAMPH,
7:VCLAMPL,8:ICLAMPH,9:ICAMPLL

Definition

```
void qspi_set_dac_offset(int bdno,int cno, int pno,int dac,int rv);
```

Description

set DAC offset to rv for board bdno pin pno,if bdno=0, set all boards,if pno=0, set all pins
offset and gain are for calibration.
dac=1..10 if dac=0 set all DAC
1:VTT,2:VIL,3:VOH,4:VOL,5:FV/VIH,6:FI,,7:VCLAMPH,
8:VCLAMPL,9:ICLAMPH,10:ICAMPLL



Definition

int qspi_get_dac_offset(int bdno,int cno, int pno,int dac);

Description

get DAC offset from board bdno pin pno

dac=1..10

1:VTT,2:VIL,3:VOH,4:VOL,5:FV/VIH,6:FI,7:VCLAMPH,
8:VCLAMPL,9:ICLAMPH,10:ICAMPLL

Definition

void qspi_set_dac_gain(int bdno, int cno,int pno,int dac,int rv);

Description

set DAC gain to rv for board bdno pin pno,if bdno=0, set all boards,if pno=0, set all pins

dac=1..10 if dac=0 set all DAC

1:VTT,2:VIL,3:VOH,4:VOL,5:FV/VIH,6:FI,7:VCLAMPH,
8:VCLAMPL,9:ICLAMPH,10:ICAMPLL

Definition

int qspi_get_dac_gain(int bdno,int cno, int pno,int dac);

Description

get DAC gain from board bdno pin pno

dac=1..10

1:VTT,2:VIL,3:VOH,4:VOL,5:FV/VIH,6:FI,7:VCLAMPH,
8:VCLAMPL,9:ICLAMPH,10:ICAMPLL

Definition

double qspi_cal_mv(int bdno, int cno,int pno,double CALL,double CALH,double MEASL,double MEASH);

Return Value

return MV offset

Description

calibrate MV offset/gain and MI offset accuracy for board bdno pin pno

cno=1..4

pno=1..8

CALL: Low calibration voltage, 0.5V is recommended

CALH: High calibration voltage, 8.5V is recommended

MEASL : enter the DMM measure value set by qspi_pmufv(bdno,cno,pno,CALL,2.0,-2.0)

MEASH : enter the DMM measure value set by qspi_pmufv(bdno,cno,pno,CALH,2.0,-2.0)

when running calibration, the DUTGND must connect to GND, SENSEn connect to DPSn



Definition

int qspi_cal(int bdno, int cno, int pno, int dac, double CALL, double CALH);

Return Value

return 0 when calibration success
return -1 when input erro
return -2 when ADC VREF3.0 check failed
return >0 when caliration failed

Description

Run qspi_cal_mv before run this calibration to get good accuracy.
calibrate DAC:dac accuracy to acc for board bdno pin pno
if bdno=0 all boards
cno=1..4 if cno=0 all sites
pno=1..8 if pno=0 all pins
dac=1..10 if dac= 0 all dacs
when running calibration, the DUTGND must connect to GND, SENSEn connecto DPSn
1:VTT,2:VIL,3:VOH,4:VOL,5:FV/VIH,6:FI,7:VCLAMPH,
8:VCLAMPL,9:ICLAMPH,10:ICAMPLL
CALL: Low calibration voltage, 0.5V is recommaned
CALH: High calibration voltage, 8.5V is recommaned

Definition

long qspi_cal_save(int bdno, const char* calfile);

Return Value

0 if ok

Description

save calibration data to cal file at bdno .
For calfile string, use "/" instead of "\".
Example : use "[c://OpenATE//T1.cal](#)" for "c:\OpenATE\T1.cal" .

Definition

long qspi_cal_load(int bdno, const char* calfile);

Return Value

0 if ok

Description

load calibration data from calfile to qspi board at board no bdno .
For calfile string, use "/" instead of "\".
Example : use "[c://OpenATE//T1.cal](#)" for "c:\OpenATE\T1.cal" .

Definition

void qspi_cal_reset(int bdno);

Description

reset calibration data and value to zero at board no bdno .



Definition

long qspi_cal_save_auto(int bdno,char* caldir);

Return Value

0 if ok

Description

save calibration data to cal file at bdno with file name:"qspi_XXXX.cal" in DIR:caldir.

XXXX is sno read from qspi_rd_pesno(int bdno) in Hex format.

If bdno==0, save all boards.

For calfile string, use "/" instead of "\".

Example : use "[c://OpenATE//](#)" for "c:\OpenATE\" .

Definition

long qspi_cal_load_auto(int bdno,char* caldir);

Return Value

0 if ok

Description

load calibration data from calfile to qspi board at board no bdno with file name:"qspi_XXXX.cal" in DIR:caldir.

"XXXX" is sno read from qspi_rd_pesno(int bdno) in Hex format.

If bdno==0, load all boards.

For calfile string, use "/" instead of "\".

Example : use "[c://OpenATE//](#)" for "c:\OpenATE\" .



3.5 Counter Function

Definition

```
void qspi_counter_ctp(int bdno,long data);
```

Description

set CTP to data for board bdno,if bdno=0, set all boards, ctp is 32 bit counter . ctp unit:10ns
CTP is used to set the counting period for counter.

Definition

```
void qspi_counter_start(int bdno,int onoff);
```

Description

start counter . for board bdno,if bdno=0, set all boards

Definition

```
void qspi_counter_select_ch(int bdno,int ch);
```

Description

select counter input from ch for board bdno, ch = 1..8 ,if bdno=0, set all boards
the selected ch CMPH will trigger the counter after counter start counting.

Definition

```
unsigned int qspi_counter_rd(int bdno,int cno);
```

Description

return counts(32 bit) from cno counter for board bdno after counter completes counting.



3.6 I2C Function

Definition

```
void qspi_set_i2cmode(int bdno, int onoff);
```

Description

set I2C mode for board bdno, if bdno=0, set all boards
if onoff=1, I2C mode enabled. Pin 1 function changed to I2C SCK(output), pin1 changed to SDA (inout).
This function will override SPI mode.

Definition

```
void qspi_i2c_itp(int bdno, long data);
```

Description

set ITP to data for board bdno, if bdno=0, set all boards, itp is 8 bit counter. ITP unit: 10ns
ITP is used to set the I2C clk period. The I2C clk period is 8 X ITP.
Example for 400K clk speed ITP=set to 31.25~= 32

Definition

```
int qspi_i2c_wrbytes(int bdno, int cno, int dev, int add, int data, int bcnt);
```

Description

write add/data to cno I2C at dev on board bdno. cno=1..4, if cno=0, set all I2Cs, if bdno=0, set all boards,
dev: 7 bits I2C address
add: 8 bits first byte to write
data: 8 bits second byte to write to internal address
bcnt: byte count 1..4
return 0 when operation success

Definition

```
int qspi_i2c_rbytes(int bdno, int cno, int dev, int add, int bcnt);
```

Description

operate I2C read bytes from cno I2C for board bdno cno=1..4, if cno=0, set all I2Cs, if bdno=0, set all boards
dev: 7 bits I2C address
add: 8 bits first byte to write to internal address
return 0 when operation success

Definition

```
int qspi_i2c_getwords(int bdno, int cno, int wcnt);
```

Description

operate I2C read bytes from cno I2C for board bdno cno=1..4
return read data after successful qspi_i2c_rbytes operation.
Wcnt: 1..2 indicate words to read. 1: byte 1..4, 2: byte 5..8



3.7 SPI Function

Definition

```
void qspi_set_spimode(int bdno, int onoff);
```

Description

set SPI mode for board bdno, if bdno=0, set all boards

if onoff=1, SPI mode enabled. Pin 3 function changed to SPI SCK(output), pin4 changed to MOSI (output). Pin5 changed to MISO(input).

This function will override I2C mode.

Definition

```
void qspi_spi_mode(int bdno, int smode);
```

Description

select SPI mode controll bits CPHA,CPOL for board bdno, if bdno=0, set all boards

smode=0 : CPHA=0,CPOL=0

smode=1 : CPHA=1,CPOL=0

smode=2 : CPHA=0,CPOL=1

smode=3 : CPHA=1,CPOL=1

Definition

```
void qspi_spi_stp(int bdno, long data);
```

Description

set STP to data for board bdno, if bdno=0, set all boards, STP is 16 bit counter . STP unit: 10ns

STP is used to set the SPI clk period . The SPI clk period is STP.

Example for 1MHz clk speed STP=100

Definition

```
void qspi_spi_wrbytes(int bdno, int cno, int data, int bcnt);
```

Description

read/write data to SPI for board bdno chip cno , if bdno=0, set all boards ,if cno=0 set all chips

bcnt= 2..32 (bit count to read/write)



Definition

int qspi_spi_getword(int bdno, int cno);

Description

return data read by qspi_spi_wrbytes from board bdno chip cno. Cno=1..4

Definition

int qspi_rd_peid(int bdno);

Description

Read 16 bits peid(daughter board) from user io port .

Definition

int qspi_rd_pesno(int bdno);

Description

Read 16 bits pesno(daughter board serial number) .



4 Pattern Format

Basic Commands:

1.FC counts :Repeat pattern counts times and go to next pattern.

#Note:counts is from 1~1048575(20 BIT)

Data Types

1. '1' :Drive 1
2. '0' :Drive 0
3. 'H' :Expect 1
4. 'L' :Expect 1
5. 'X' :Tri-state out
6. 'Z' :Tri-state in(< VOH, > VOL)
7. 'J' :Drive 1 Expect 1
8. 'Q' : Drive 0 Expect 0

Examples:

```
//  
//  
FC 1023 011000HLHHH1100XX;//REPEAT 1024  
FC 2 1110111XLL1100XX;//REPEAT 2  
FC 1 11HHLLL00000; //REPEAT 1  
FC 1 0000111111HHHLLLLL;//REPEAT 1  
FC 6 HHHHLLXXXX11110000;//REPEAT 6  
FC 1 1111100101HHHLLLLL1L1L;REPEAT 1
```



5 Pin Out

PIN OUT

| | Per SITE | qspi |
|--------|------------------|-------|
| DOUT1 | SITE1/PIN1/SLC1 | J1-65 |
| DOUT2 | SITE1/PIN2/SDA1 | J1-31 |
| DOUT3 | SITE1/PIN3/SCK1 | J1-63 |
| DOUT4 | SITE1/PIN4/MOSI1 | J1-29 |
| DOUT5 | SITE1/PIN5/MISO1 | J1-61 |
| DOUT6 | SITE1/PIN6 | J1-27 |
| DOUT7 | SITE1/PIN7 | J1-59 |
| DOUT8 | SITE1/PIN8/DPS1 | J1-25 |
| DOUT9 | SITE2/PIN1/SLC2 | J1-57 |
| DOUT10 | SITE2/PIN2/SDA2 | J1-23 |
| DOUT11 | SITE2/PIN3/SCK2 | J1-55 |
| DOUT12 | SITE2/PIN4/MOSI2 | J1-21 |
| DOUT13 | SITE2/PIN5/MISO2 | J1-53 |
| DOUT14 | SITE2/PIN6 | J1-19 |
| DOUT15 | SITE2/PIN7 | J1-51 |
| DOUT16 | SITE2/PIN8/DPS2 | J1-17 |
| DOUT17 | SITE3/PIN1/SLC3 | J1-49 |
| DOUT18 | SITE3/PIN2/SDA3 | J1-15 |
| DOUT19 | SITE3/PIN3/SCK3 | J1-47 |
| DOUT20 | SITE3/PIN4/MOSI3 | J1-13 |
| DOUT21 | SITE3/PIN5/MISO3 | J1-45 |
| DOUT22 | SITE3/PIN6 | J1-11 |
| DOUT23 | SITE3/PIN7 | J1-43 |
| DOUT24 | SITE3/PIN8/DPS3 | J1-9 |
| DOUT25 | SITE4/PIN1/SLC4 | J1-41 |



| | | |
|-----------|------------------|--|
| DOUT26 | SITE4/PIN2/SDA4 | J1-7 |
| DOUT27 | SITE4/PIN3/SCK4 | J1-39 |
| DOUT28 | SITE4/PIN4/MOSI4 | J1-5 |
| DOUT29 | SITE4/PIN5/MISO4 | J1-37 |
| DOUT30 | SITE4/PIN6 | J1-3 |
| DOUT31 | SITE4/PIN7 | J1-35 |
| DOUT32 | SITE4/PIN8/DPS4 | J1-1 |
| SENSE1 | DPS1 sense | J1-26 |
| SENSE2 | DPS2 sense | J1-18 |
| SENSE3 | DPS3 sense | J1-10 |
| SENSE4 | DPS4 sense | J1-2 |
| GND | | J1- 2,4,6,10,12,14,16,18 ,20,22,24,28,32,34,3 6,38,40,42,44,46,48, 50,52,54,56,58,62,6 6,68 |
| EXT_FORCE | | J1-64(V2.0) |
| EXT_SENSE | | J1-30(V2.0) |
| | | |



Figure1. QSPI Connector Pin out

| | | | |
|-----------|----|----|-----------|
| DOUT32 | 1 | 35 | DOUT31 |
| SENSE4 | 2 | 36 | GND |
| DOUT30 | 3 | 37 | DOUT29 |
| GND | 4 | 38 | GND |
| DOUT28 | 5 | 39 | DOUT27 |
| GND | 6 | 40 | GND |
| DOUT26 | 7 | 41 | DOUT25 |
| | 8 | 42 | GND |
| DOUT24 | 9 | 43 | DOUT23 |
| SENSE3 | 10 | 44 | GND |
| DOUT22 | 11 | 45 | DOUT21 |
| GND | 12 | 46 | GND |
| DOUT20 | 13 | 47 | DOUT19 |
| GND | 14 | 48 | GND |
| DOUT18 | 15 | 49 | DOUT17 |
| GND | 16 | 50 | GND |
| DOUT16 | 17 | 51 | DOUT15 |
| SENSE2 | 18 | 52 | GND |
| DOUT14 | 19 | 53 | DOUT13 |
| GND | 20 | 54 | GND |
| DOUT12 | 21 | 55 | DOUT11 |
| GND | 22 | 56 | GND |
| DOUT10 | 23 | 57 | DOUT9 |
| GND | 24 | 58 | GND |
| DOUT8 | 25 | 59 | DOUT7 |
| SENSE1 | 26 | 60 | |
| DOUT6 | 27 | 61 | DOUT5 |
| GND | 28 | 62 | GND |
| DOUT4 | 29 | 63 | DOUT3 |
| EXT-SENSE | 30 | 64 | EXT-FORCE |
| DOUT2 | 31 | 65 | DOUT1 |
| GND | 32 | 66 | GND |
| DUTGND | 33 | 67 | |
| GND | 34 | 68 | GND |